

Plutus Bench

Niels Mündler, OpShin

March 17, 2024

1 Introduction

1.1 Background

In an era where decentralized technologies are rapidly evolving, the development of robust and efficient smart contracts has never been more critical. Plutus Bench thrives to become a solid foundation within the Cardano blockchain ecosystem, aiming to streamline the development process for smart contracts. This report marks the completion of the first milestone in the ambitious journey of Plutus Bench, laying down the foundational stones of project inception and team assembly, alongside extensive market research and community engagement.

1.2 Purpose of this Document

This document serves as a comprehensive report on the first milestone of Plutus Bench. It outlines the project's inception, team assembly, market research, and community feedback, providing a detailed insight into the project's vision and the requirements for the Plutus Bench tool. Furthermore, it establishes a direct line of communication with the community, ensuring that Plutus Bench evolves as a community-centric project.

1.3 Scope

This document is structured as follows:

- **Project Inception:** This section provides an overview of the project's inception, outlining the vision and objectives of Plutus Bench.
- **Team Assembly:** This section introduces the team behind Plutus Bench, highlighting the collective expertise and passion that drives the project forward.
- **Market Research:** This section delves into the market research conducted by the Plutus Bench team, providing insights into the demands and challenges faced by developers working with Cardano smart contracts
- **Planned features:** This section presents the detailed requirements for Plutus Bench, guided by community insights and market analysis, setting a clear path forward for the project.

2 Project Inception

The inception of Plutus Bench is rooted in the vision of streamlining the development process for smart contracts on the Cardano blockchain. The project aims to provide a comprehensive toolset that empowers developers to build, test, and deploy smart contracts with ease and efficiency. At the heart of this vision is the commitment to address the specific challenges and requirements faced by developers working with Cardano smart contracts, ensuring that Plutus Bench is not just a tool but a solution tailored to the needs of the community. The main objectives of Plutus Bench are as follows:

- **Streamline Benchmarking:** Plutus Bench seeks to streamline the process of benchmarking smart contracts, providing tools and utilities that automate the

benchmarking process and provide comprehensive insights into the performance of smart contracts.

- **Community-Centric Approach:** The project is committed to a community centric approach, ensuring that the tools and utilities provided by Plutus Bench are tailored to the specific demands and challenges faced by developers working with Cardano smart contracts.

3 Team Assembly

At the heart of this milestone is the formation of a dynamic team, brought together by a shared vision and a collective expertise in Python, blockchain technologies, and specifically, the Cardano smart contract environment. This team is not just a group of individuals but a synergy of skills, passion, and dedication towards making Plutus Bench a reality.

Due to the widespread usage of the Python programming language, we are able to source developers from outside the community to work on the project. This is a significant advantage, as it allows us to tap into a larger pool of talent and expertise, ensuring that Plutus Bench is developed with the highest standards of quality and efficiency.

The team is led by Niels Mündler, a seasoned Python developer with a deep understanding of blockchain technologies, in particular the UPLC foundations underlying Cardano Smart Contracts.

As a central point of contact, Niels is responsible for steering the project towards its objectives, ensuring that the team remains focused and motivated throughout the development process. He can be reached through the specifically created email address plutus-bench@opshin.dev.

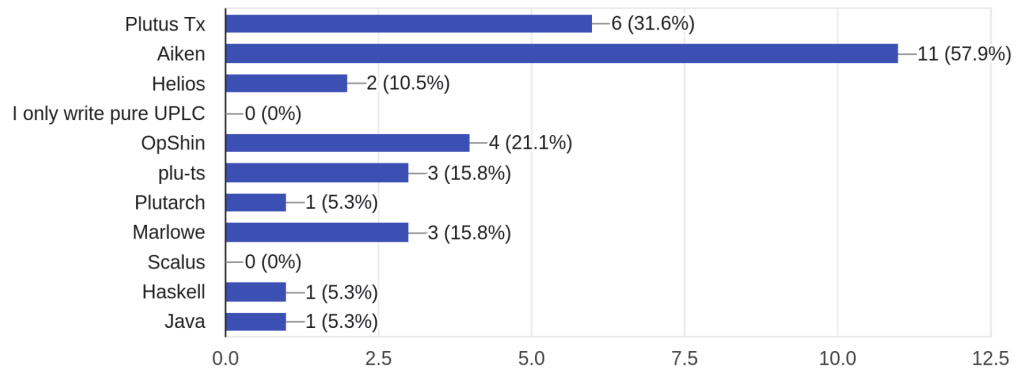
4 Market Research

To assess the current needs of the community, a comprehensive market research was conducted by the Plutus Bench team. The research aimed to identify the specific challenges and requirements faced by developers working with Cardano smart contracts, providing insights that would shape the vision and requirements of Plutus Bench. The research was conducted through a combination of a survey on a variety of development forums, and direct engagement with the Cardano community, ensuring that the insights were drawn from a diverse range of perspectives and experiences. The raw results of the survey are published on the Plutus Bench GitHub repository¹. The key insights drawn from the market research are as follows:

¹https://github.com/OpShin/plutus-bench/tree/master/report/survey_results.csv

What Smart Contract languages do you normally use?

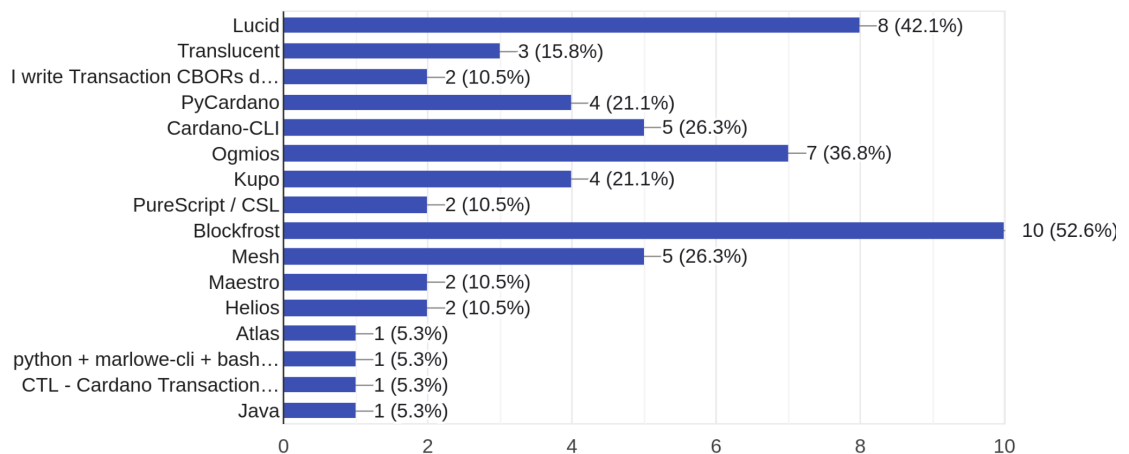
19 responses



Language diversity: Developers use a variety of languages that compile to Plutus Core, including Haskell, Python, and JavaScript. Although aiken emerges as the most popular language with 60% of users claiming to use it regularly, there is a clear demand for language-agnostic tools that support smart contracts written in any language.

What tools do you use to write off-chain code?

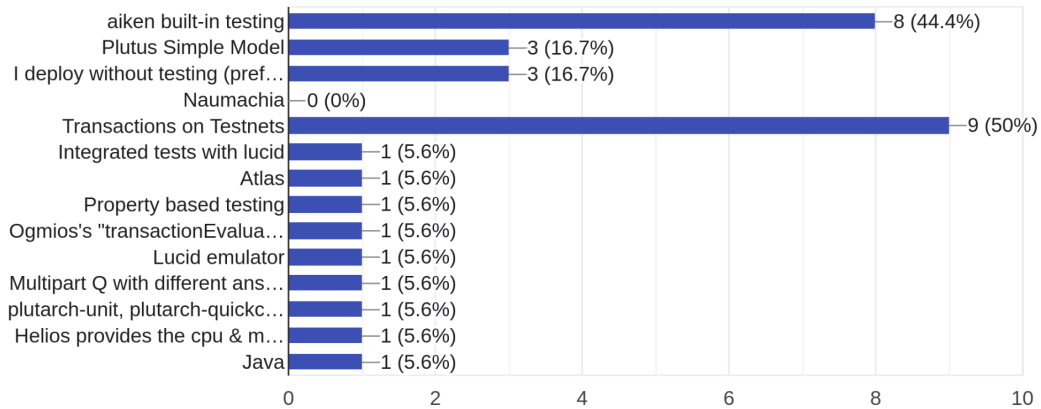
19 responses



Reliance on Blockfrost API: For building off-chain components, developers rely heavily on the Blockfrost API. This indicates a need for tools that can emulate the Blockfrost API locally, providing a seamless integration of existing off-chain components with Plutus Bench.

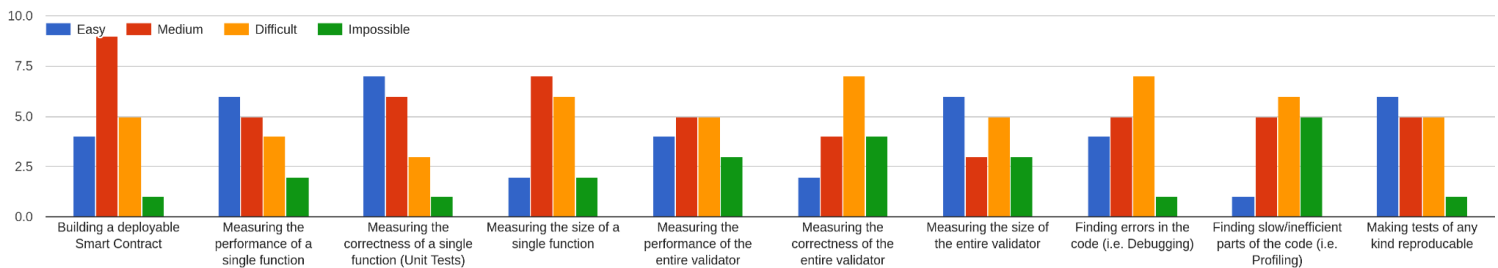
What tools do you use to assess the quality of on-chain code?

18 responses



Lack of dedicated testing for Smart Contracts: The survey revealed that developers face challenges in testing smart contracts. While the newly developed built-in unit testing in aiken is used by 44% of survey participants, 50% rely on manually submitting transactions to testnets and the remaining answers describe many custom solutions. This indicates a clear demand for a dedicated testing framework that streamlines the process of testing smart contracts. Note that the aiken testing framework does not yet support testing of full smart contract transactions, only functions thereof.

Please rate your experience with these tasks



1. **Difficulty in testing smart contract correctness:** The survey revealed that developers face challenges in testing the correctness of smart contracts. Most describe performing assessments of the correctness of a validator "Difficult" with 4 participants even describing it "Impossible". Similarly, assessing smart contract performance is widely seen as "Medium" to "Difficult" with 3 participants describing it "Impossible". This indicates a clear demand for tools that automate the process of testing the correctness and performance of smart contracts.
2. **Difficulty of line-level assessments:** Most developers describe the process of debugging and profiling smart contracts "Medium" to "Difficult". This indicates a

clear demand for tools that provide line-level insights into the smart contracts, streamlining the process of debugging and profiling.

5 Planned Features

The insights drawn from the market research and community feedback have shaped the vision and requirements of Plutus Bench, ensuring that the project is tailored to the specific demands and challenges faced by developers working with Cardano smart contracts. The planned features of Plutus Bench are as follows:

- **Dedicated Transaction Testing:** Plutus Bench will provide a dedicated testing framework that streamlines the process of testing smart contracts transactions, ensuring that developers can test smart contracts with ease and efficiency. In particular, we plan to provide a simple endpoint that locally evaluates constructed transactions involving smart contracts and returns the result or error logs and performance measurements.
- **Blockfrost API Emulation:** Plutus Bench will provide tools that emulate the Blockfrost API locally, ensuring a seamless integration of existing off-chain components with the Plutus Bench toolset. Note that this implies agnostic support for any off-chain component.
- **Language Agnostic Design:** Plutus Bench will be designed to be language agnostic, ensuring that developers can build, test, and measure smart contracts written in any language that compiles to Plutus Core.
- **Line-Level Insights:** Plutus Bench will provide tools that enable line-level insights into the smart contracts, streamlining the process of debugging and profiling. The prototype will execute source Python code in a debugger upon smart contract execution, providing line-level insights into the smart contracts and the ability to debug the code by stepping through lines.

6 Conclusion

The completion of the first milestone marks the inception of Plutus Bench, laying down the foundational stones of project inception and team assembly, alongside extensive market research and community engagement. The insights drawn from the market research and community feedback have shaped the vision and requirements of Plutus Bench, ensuring that the project is tailored to the specific demands and challenges faced by developers working with Cardano smart contracts. The planned features of Plutus Bench are designed to streamline the development process for smart contracts on the Cardano blockchain, providing a comprehensive toolset that empowers developers to build, test, and deploy smart contracts with ease and efficiency.